На правах рукописи

Чалый Дмитрий Юрьевич

МОДЕЛИРОВАНИЕ И АНАЛИЗ СЕТЕВЫХ ТРАНСПОРТНЫХ ПРОТОКОЛОВ С ПОМОЩЬЮ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ

01.01.09 — дискретная математика и математическая кибернетика

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Ярославль — 2006

Работа выполнена в Ярославском государственном университете им. П.Г. Демидова на кафедре теоретической информатики.

Научный руководитель –	кандидат физико-математических наук, профессор Соколов Валерий Анатольевич
Официальные оппоненты:	доктор физико-математических наук, профессор Ломазова Ирина Александровна
	кандидат физико-математических наук, Непомнящий Валерий Александрович
Ведущая организация –	Институт системного программирования РАН

Защита состоится "____"_____ 2006 в ____ ч. ____ мин. на заседании диссертационного совета Д 212.002.03 при Ярославском государственном университете им. П.Г. Демидова по адресу: 150008, г. Ярославль, ул. Союзная 144.

С диссертацией можно ознакомиться в библиотеке ЯрГУ им. П.Г. Демидова.

Автореферат разослан "____"_____ 2006.

Учёный секретарь диссертационного совета Яблокова С.И.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Объект исследования и актуальность темы. Одним из важнейших достижений научно-технического прогресса в настоящее время являются коммуникационные системы, представляющие собой сети передачи информации. Успешное внедрение Интернет и интранет технологий приводит к тому, что человечество становится все более зависимым от надежности функционирования вычислительных устройств.

Транспортные протоколы являются важным элементом коммуникационной архитектуры сети Интернет (согласно исследованиям К. Томпсона, Дж. Миллера и Р. Уайлдера, около 95% всех переданных байтов и 85-95% всех переданных пакетов). Основной задачей протокола транспортного уровня является предоставление сервиса программным процессам для надежного и эффективного обмена информацией через ненадежную среду передачи — коммуникационную сеть. С точки зрения транспортного протокола сеть представляется в виде «черного ящика», где информация может теряться, переупорядочиваться, искажаться и, возможно, дублироваться. Под эффективностью работы транспортного протокола понимается прежде всего эффективное использование сетевых ресурсов — таких как, например, пропускная способность каналов передачи и буферов маршрутизаторов.

Предметом нашего исследования является транспортный протокол TCP (Transmission Control Protocol), который является основным транспортным протоколом коммуникационной архитектуры сети Интернет. Так как этот протокол постоянно изменяется и дополняется, то можно говорить о семействе протоколов TCP.

Исследование свойств транспортных протоколов, в частности различных версий протокола TCP, является важной и актуальной задачей, которая рассматривалась в ряде работ. Основным объектом исследований являлись алгоритмы управления потоком транспортных протоколов, а основным методом исследований, который использовался в этих работах — имитационное моделирование. Яркими представителями таких работ являются работы К. Фолла и С. Флойд по исследованию различных версий алгоритма управления потоком протокола TCP; в работах

С. Шао, М. Санадиди и М. Герла, которые представляют новый протокол TCP Westwood, и обосновывают его эффективность; работы И. В. Алексеева и В. А. Соколова, которые представляют новый протокол ARTCP (Adaptive Rate TCP) и на ряде модельных экспериментов обосновывают его преимущество перед стандартным TCP. Другим популярным методом исследования является построение аналитических моделей. Например, в работах У. Виллингера и М. Таггу рассматриваются модели алгоритма управления потоком и вопросы порождения протоколом TCP самоподобного трафика. Методы имитационного моделирования могут быть весьма экономичными для выявления многих ошибок, однако проверить все возможные взаимодействия и пути выполнения протокола вряд ли представляется возможным.

Одним из подходов к решению задачи корректности транспортных протоколов может быть построение формальных моделей и их последующий анализ с помощью формальных методов (например, методов model checking). Вопросы корректности коммуникационных протоколов рассматривались, например, в работах Дж. Хольцмана, который предлагает строить модели протоколов с помощью конечных автоматов и использовать эти модели для последующей верификации; в работах В. А. Непомнящего, Т. Г. Чуриной и Е. В. Окунишниковой предлагаются методы по моделированию спецификаций, представленных на языках Estelle и SDL с помощью сетей Петри высокого уровня. Однако применение этих подходов к моделированию и верификации семейства протоколов TCP затруднено тем, что стандартные документы, задающие их спецификацию, изложены на неформальном языке. Были опубликованы работы по построению моделей протокола TCP в терминах раскрашенных сетей Петри. В работе Х. Фигейредо и Л. Кристенсена представляются результаты моделирования процесса обмена данными ряда версий протокола TCP, а в работе Б. Хана и Дж. Биллингтона рассматривается модель процессов установки и завершения соединений. Эти работы рассматривают некоторые фрагменты оригинального протокола; перед нами же стояла задача промоделировать стандарт протокола целиком. Другой особенностью этих работ является то, что основной акцент они делают именно на

построении моделей, а не на разработке методов их исследования. В нашей работе кроме задачи построения моделей, рассматриваются вопросы анализа различных свойств протоколов, таких как производительность и корректность.

Цели и задачи работы. Создание новых модификаций протокола TCP, которые являются более эффективными и/или корректными, является несомненно важной задачей. При этом корректность исполнения протокола имеет приоритет перед производительностью. Поэтому создание формальных моделей протоколов и разработка способов анализа этих моделей является одним из методов, которые позволяют верифицировать корректность и убедиться в эффективности работы протокола.

Таким образом, главными целями проведенной работы являются:

1. разработка формальной технологии моделирования спецификаций семейства транспортных протоколов TCP;
2. описание и применение формальных методов анализа свойств полученных моделей;
3. создание эффективных алгоритмов работы транспортных протоколов.

Для достижения этих целей были поставлены следующие задачи:

1. разработать формальную модель последней версии стандартной спецификации протокола TCP;
2. описать методы модификации модели для представления новых версий этого протокола;
3. с помощью формального математического аппарата провести верификацию свойств построенной модели;
4. разработать алгоритмы более эффективного восстановления от множественных потерь для транспортного протокола ARTCP;
5. провести оценку эффективности приведенных алгоритмов для протокола ARTCP по сравнению со протоколами семейства TCP.

Методы исследования и формальный аппарат. На различных этапах работы применялись различные методы исследования. На начальном этапе моделирования проводился анализ стандартной спецификации протокола TCP и документов, которые описывают протокол ARTCP. Далее с помощью формализма раскрашенных сетей Петри строилась формальная модель транспортных протоколов, и разрабатывались методы ее модификации для представления новых версий протоколов. Верификация построенной модели проводилась с помощью метода model checking с использованием темпоральной логики и методов анализа множеств достижимых состояний, разработанных для формализма раскрашенных сетей Петри. Для исследования производительности рассматриваемых транспортных протоколов использовался метод имитационного моделирования.

Научная новизна. Научной новизной обладают следующие решения, выносимые на защиту:

1. построенная модель последней стандартной спецификации протокола TCP и разработанная технология модификации этой модели для представления новых версий протокола;
2. с помощью представленной технологии была разработана формальная модель протокола ARTCP;
3. с использованием методов model checking был проверен ряд режимов работы протокола и было показано, что модель соответствует стандартной спецификации протокола;
4. в связи с обновлением спецификации протокола TCP, усовершенствована спецификация протокола ARTCP и описаны эффективные алгоритмы для восстановления множественных потерь сегментов;
5. приведены результаты модельных экспериментов, показывающие эффективность разработанных алгоритмов для восстановления от множественных потерь для протокола ARTCP.

Практическое значение работы. Разработанная технология моделирования была реализована в системе моделирования Design/CPN и

может быть применена для автоматизированного анализа новых транспортных протоколов семейства TCP.

Апробация работы. Результаты работы докладывались на международной конференции Parallel Computer Technologies (Нижний Новгород, 2003 год), Всероссийской научной конференции «Методы и средства обработки информации» (Москва, 2003 год), международной конференции по вычислительной математике (Новосибирск, 2004 год), междисциплинарной конференции НБАТТ-21 (Петрозаводск, 2004 год), XVI Международной научно-технической конференции «Математические методы и информационные технологии в экономике, социологии и образовании» (Пенза, 2005 год). Кроме того, полученные результаты обсуждались на семинаре «Моделирование и анализ информационных систем» ЯрГУ. Материалы диссертации вошли в отчет по гранту РФФИ №03-01-00804.

Публикации. За время работы над диссертацией было опубликовано 7 публикаций.

Структура и объем работы. Диссертация состоит из введения, трех глав и заключения. Работа изложена на 148 страницах, иллюстрирована 46 рисунками и содержит 10 таблиц. Список литературы содержит 102 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Введение

Во введении обосновывается актуальность выбранной тематики исследований и формулируются поставленные задачи, характеризуются научная новизна и практическая ценность работы.

Глава 1. Коммуникационные транспортные протоколы

Первая глава посвящена описанию транспортных протоколов коммуникационной архитектуры TCP/IP. Здесь также подробно рассмотрен объект исследования — коммуникационный транспортный протокол TCP и его модификации.

В разделах 1.1-1.3 приводится современный взгляд на сети передачи данных. Упомянуты основные составляющие коммуникационной сети (раздел 1.1), приведена стандартная модель взаимодействия открытых

систем (раздел 1.2) и обзорно рассмотрено назначение различных протоколов стека TCP/IP (раздел 1.3).

Протокол TCP (Transmission Control Protocol — протокол управления передачей) является основным протоколом транспортного уровня, используемым в Internet. С момента своего появления в 1981 году, опубликованный в RFC793 (RFC — Request for Comments), он постоянно совершенствовался и на данный момент существует множество документов, регламентирующих различные аспекты его работы. Раздел 1.4 посвящен описанию этого транспортного протокола и его модификаций. Пользователями сервиса, предоставляемого протоколом TCP, являются пользовательские программные процессы. Один из вариантов интерфейса между пользовательскими процессами и TCP рассмотрен в документе RFC793. Существуют альтернативные подобные интерфейсы, например, интерфейс Berkeley Sockets. Мы будем рассматривать только интерфейс, изначально описанный в спецификации протокола RFC793, и приведенный в п. 1.4.1 работы.

Взаимодействие протокола TCP с протоколом нижележащего уровня описано в п. 1.4.2. Протокол TCP использует сервис, предоставляемый протоколом IP, для обеспечения коммуникации с удаленной стороной соединения. Протокол TCP разбивает полученные от пользователя данные на части и «упаковывает» их в специальную структуру, называемую сегментом. С помощью сегментов протокол отправляет не только данные, но и различные служебные сообщения. Протокол TCP использует протокол IP для отправления сегментов в сеть и получения их из сети.

В п. 1.4.3 рассматривается управление транспортным соединением. Во время работы протокол TCP управляет логическим соединением между процессами на конечных системах коммуникационной сети, где создаются локальные соединения, соответствующие взаимодействующим программным процессам. Чтобы поддерживать локальное соединение в рабочем состоянии протокол должен знать значения ряда параметров, которые хранятся в специальной структуре, называемой контрольном блоке соединения. Далее в работе описываются изменения этих параметров при стандартных фазах работы транспортного соединения — открытии и

закрытии соединения.

В п. 1.4.4 рассматриваются алгоритмы работы протокола, которые обеспечивают надежную передачу данных через ненадежную среду связи. TCP-соединения являются полнодуплексными, т.е. информация может передаваться одновременно в обоих направлениях. Каждому передаваемому байту присваивается уникальный номер возрастающей последовательности передачи. Принятые получателем сегменты помещаются в буфер, из которого уже передаются программному процессу. Кроме этого получатель обязан сообщать отправителю с помощью подтверждений о тех байтах, которые уже получены. Подтверждения имеют кумулятивный характер, т.е. если отправитель получил подтверждение с номером n, то это значит что все байты по $(n-1)$-й включительно успешно получены. В случае неполучения подтверждения в течение некоторого времени, протокол производит повторную передачу по тайм-ауту тех сегментов, которые он считает потерянными. Все эти механизмы позволяют протоколу TCP предоставлять надежный сервис по передаче данных через ненадежную среду, которая может терять, искажать (искажение сегмента расценивается как потеря), дублировать или переупорядочивать переданную информацию.

Для управления передачей сегментов протокол использует два вида алгоритмов. Во-первых, протоколом используется алгоритм «скользящего окна» (sliding window algorithm), для того чтобы предотвратить перегрузку буфера получателя. В процессе освобождения буфера получатель обязан сообщать об этом удаленной стороне. Во-вторых, для того чтобы реагировать на загруженность коммуникационной сети, используются алгоритмы управления потоком, стандарт которых представлен в RFC2581, RFC3390 и RFC3782.

Далее в работе описываются различные расширения протокола TCP, которые направлены на улучшение эффективности и надежности его работы. Рассматриваются мехнизмы выборочных подтверждений (selective acknowledgements), масштабирования окна (window scaling), алгоритмы, использующие временную метку (timestamp), алгоритм Limited Transmit.

После этого рассматривается алгоритм ARTCP (Adaptive Rate TCP), который характеризуется рядом существенных отличий от стандартного алгоритма управления потоком TCP. Так, скорость отправки сегментов в сеть определяется не размером окна передачи (как в TCP), а индивидуальной задержкой каждого сегмента. Изменение скорости передачи потока выражается в изменении его скважности (межсегментного временного интервала). Индикатором текущего состояния сети служит изменение скважности потока, измеряемое получателем, а не потеря сегмента как в стандартном TCP. В протоколе ARTCP устранена логическая зависимость алгоритмов коррекции ошибок передачи и управления потоком.

Заканчивается глава обзором основных методов моделирования рассмотренного семейства транспортных протоколов и вопросами моделирования порождаемого этими протоколами трафика.

Глава 2. Моделирование транспортных протоколов с помощью раскрашенных сетей Петри.

В главе представляется построенная модель семейства транспортных протоколов TCP и рассматриваются принципы модификации модели для моделирования последующих версий протокола TCP.

В разделе 2.1 рассматривается определение структуры раскрашенных сетей Петри (CP-сети), их динамическое поведение, расширения этого класса сетей Петри, а также методы анализа свойств моделей и автоматические средства, поддерживающие этот формализм. Раскрашенная сеть Петри представляет собой ориентированный граф с двумя видами вершин — позициями и переходами, а также имеет блок определений. Блок определений раскрашенной сети Петри содержит определения типов, операций и функций, которые могут быть использованы для задания выражений, использующихся в CP-сети. На практике для этих целей используется язык CPN ML.

С каждой позицией сети связан тип, из которого могут принимать значения маркеры в данной позиции. Позиции соединяются с переходами с помощью конечного множества дуг. С помощью переходов моделируется динамика, или действия, изменяющие разметку позиций. Каждый пе-

реход имеет охранное выражение, которое задает условия срабатывания перехода. Срабатывание перехода изымает из каждой входной позиции маркеры, согласно выражениям на входных дугах перехода, и помещает мультимножество маркеров в позиции согласно выражениям на выходных дугах перехода.

Раскрашенные сети Петри могут быть расширены концепцией времени. При этом к модели добавляется монотонно возрастающий счетчик, называемый глобальными часами. Кроме этого некоторые типы маркеров объявляются временны́ми, т.е. маркеры этих типов могут кроме собственно значения, иметь специальную временну́ю метку. Если значение временной метки маркера больше, чем значение глобальных часов, то такой маркер не может участвовать в срабатываниях переходов. Значение глобальных часов не меняется, пока существуют активные переходы, которые могут сработать. Когда таких переходов нет, но они появятся, если глобальное время изменится, происходит увеличение значения глобальных часов на минимальное значение, когда станет активным хотя бы один переход.

Другим полезным расширением раскрашенных сетей Петри является введение иерархических конструкций. Иерархическая раскрашенная сеть Петри — это композиция множества неиерархических сетей. Введение иерархических конструкций позволяет разбить модель на несколько взаимодействующих подсетей.

Одним из преимуществ формализма раскрашенных сетей Петри является то, что для него разработано большое количество методов анализа различных свойств. Раскрашенная сеть Петри может представлять не только структуру моделируемой системы, но и посредством срабатывания переходов мы можем видеть, как система работает. Одним из мощных формальных методов для проведения таких исследований является построение и анализ множеств достижимых состояний, также называемых графами достижимости раскрашенных сетей Петри. При этом граф достижимых состояний содержит информацию не только о достижимых состояниях сети, но и о сработавших переходах. Исследование свойств графов достижимости позволяет проводить анализ таких свойств как до-

стижимость некоторой разметки, живость переходов и т.д.

Темпоральные логики, такие как CTL, являются мощным инструментом для исследования свойств параллельных и распределенных систем. Задачу проверки модели с помощью темпоральных логик можно сформулировать следующим образом. Пусть задана формальная модель, некоторое состояние модели и формула темпоральной логики φ. Необходимо выяснить, удовлетворяет ли это состояние модели формуле темпоральной логики φ? Для раскрашенных сетей Петри была разработана логика ASK-CTL, которая может задавать свойства не только на множестве позиций, но и на множестве переходов.

Основные концепции формализма раскрашенных сетей Петри демонстрируются на оригинальной модели классического примера — тунеле Клейтона.

Далее в главе представляется один из основных результатов диссертации — модель транспортных протоколов в терминах формализма раскрашенных сетей Петри. В разделе 2.2 приводится описание основных типов, с помощью которых моделируется состояние различных служебных структур протокола. Например, приводятся примеры описания типов, которые моделируют вызововы пользователя и ответные реакции протокола, сегменты, контрольный блок соединения, таймеры, а также параметры конечных систем коммуникационной архитектуры TCP/IP.

Раздел 2.3 посвящен описанию иерархической структуры модели. После анализа официальных документов, описывающих стандарт транспортного протокола TCP, было решено разбить модель на ряд функциональных модулей, каждый из которых моделирует определенную часть оригинального протокола. В терминах раскрашенных сетей Петри это разбиение можно представить в виде иерарахического дерева, каждый элемент которого представляет собой модель некоторой части оригинального протокола. Связи между элементами этого дерева показывают взаимосвязь подмоделей.

В разделе 2.4 рассматриваются принципы, по которым строились подсети, моделирующие отдельные аспекты работы протокола. Первый шаг в построении подмодели функциональной части состоит в том, что-

бы определить те служебные структуры протокола, которые необходимы для работы этой функциональной части протокола. Решение принимается на основе анализа стандартной спецификации протокола. После этого создаются переходы, которые моделируют действия, необходимые для работы рассматриваемого функционального аспекта протокола. В большинстве случаев создается один переход, который собственно и осуществляет моделирование. Однако в ряде случаев требуется два перехода. Это имеет смысл делать тогда, когда моделируемый аспект имеет несколько вариантов работы.

Следующим шагом позиции и переходы соединяются с помощью дуг. Если некоторая структура протокола должна быть создана в процессе работы моделируемого аспекта, то создается исходящая дуга. В том случае, если служебная структура протокола удаляется, то создается входящая дуга. Если происходит обновление значения служебной структуры, то создается пара дуг — входящая, с помощью которой изымается маркер со старым значением структуры и исходящая, с помощью которой в позицию помещается маркер с новым значением структуры.

После этого необходимо задать охраны переходов подсети, которые будут определять, когда действие может быть выполнено, и выражения на дугах, которые показывают, как будет изменяться разметка сети, а соответственно и служебные структуры протокола. В общем случае для каждого из переходов подсети нам необходимо реализовать следующие функции:

- предикат действия, который определяет, должно ли действие выполниться при текущих условиях или нет. Эта функция будет использоваться в качестве охранного выражения перехода;
- функцию, которая задает алгоритм, по которому изменяются служебные структуры протокола. В нашей модели эта функция расчитывает значение контрольного блока соединения. Значения остальных служебных структур расчитываются стандартным образом на основе получившегося значения контрольного блока соединения;
- функцию, определяющую, какие сегменты должны быть отправле-

ны удаленной стороне соединения в ответ на данное действие;

- функцию, которая определяет, какие сигналы должны быть отправлены пользовательскому процессу при совершении действия.

Конкретная реализация этих функций выводится во внешние модули на языке SML, которые подключаются к модели. После этих действий подсеть с помощью стандартных механизмов присоединяется к основной модели.

В разделе 2.5 описывается модель обработки пользовательских вызовов протоколом. В процессе анализа этого аспекта работы протокола TCP мы сделали вывод, что любой пользовательский вызов может либо открыть транспортное соединение, либо быть обработанным, либо сбросить уже существующее транспортное соединение. В работе приводятся подсети, моделирующие эти ситуации.

Раздел 2.6 посвящен модели передачи сегментов в коммуникационную сеть. Представленные подсети моделируют следующие варианты передачи: передача данных от пользовательского процесса удаленной стороне; передача подтверждений удаленной стороне соединения; различные механизмы повторной передачи (по тайм-ауту и быстрой повторной передачи); передачи сегментов в сеть с заданной скоростью (используется, например, при моделировании протокола ARTCP).

Раздел 2.7 описывает модель обработки пришедших сегментов, состоящую из подсетей, моделирующих следующие варианты обработки: обработка синхронизирующих сегментов, которые используются при установке транспортного соединения; моделирование начальную обработку сегмента, в частности проверки, выполняемые при приеме сегмента; моделирование обработки сегментов в порядке их очередности.

Каждая из представленных подсетей является достаточно компактной и содержит не более двух переходов.

С момента публикации начальной спецификации протокола в 1981 году протокол TCP постоянно изменялся. Эти изменения были направлены как на исправление ошибок в протоколе, так и на введение новых механизмов, которые позволяют протоколу работать более эффективно.

В разделе 2.8 рассмотрены методы, которые позволяют модифицировать модель для того, чтобы она могла представлять новые версии протокола TCP. Любое изменение протокола TCP затрагивает обычно изменение алгоритма работы протокола. Как следствие, иногда для этого необходимо вводить новые параметры для работы этих изменений (например, в контрольный блок соединенения). Таким образом, для модификации модели необходимо предоставить достаточно удобные средства, которые позволяют модифицировать как структуры данных оригинального протокола, так и алгоритм работы протокола.

В работе рассматриваются два основных направления модификации алгоритма протокола: введение различных расширений и изменение алгоритма управления потоком. Кроме этого приводятся общие рекомендации по модификации модели. Под расширениями протокола понимаются прежде всего алгоритмы, предложенные в RFC1323, RFC2018, RFC2883. Эти алгоритмы расширяют функциональность протокола, являются стандартизированными, однако нигде не указано, что реализация, которая соответствует стандарту протокола, обязана их включать. Решение о том, необходимо ли учитывать алгоритм работы расширения при работе локального соединения, принимается на этапе синхронизации соединения. В работе рассматривается метод модификации модели для хранения параметров работы таких расширений, а также метод модификации блока определений модели, который позволяет реализовать алгоритм работы этих расширений. В отличие от расширений, которые могут быть включены или нет в реализацию протокола, алгоритм управления потоком обязан присутствовать в транспортном протоколе. Здесь мы также рассматриваем методы модификации модели, которые позволяют моделировать хранение параметров алгоритма управления потоком и его работу.

В общем случае метод модификации модели для представления новой версии протокола можно представить в следующем виде:

1. адаптация существующих и/или определение новых структур для хранения параметров, используемых новой версией протокола;
2. определение того, какие функциональные модули протокола затра-

гивает модификация;

3. внесение изменений в эти функциональные модули и/или программный код, отвечающие за работу этих модулей;
4. расширение оригинальной модели подсетями, реализующими необходимую функциональность, не предусмотренную в представленной модели протокола.

Раскрашенные сети Петри позволяют выполнять созданные в терминах этого формализма модели, следовательно, полученные модели могут быть полезны в различных модельных экспериментах. В разделе 2.9 дается представление о том, как можно создавать сценарии этих экспериментов. Рассматриваются случаи, когда требуется моделирование работы одного или одновременно нескольких транспортных соединений.

Глава 3. Анализ свойств коммуникационных транспортных протоколов.

Представленная в предыдущей главе модель используется как основа для последующего анализа корректности важных аспектов работы транспортного протокола — процессов открытия и закрытия соединения. В настоящей главе описывается подход, который позволяет убедиться в том, что построенная модель адекватно отражает спецификацию протокола, а также приводятся результаты экспериментов.

В разделе 3.1 дается введение в проблематику верификации транспортных протоколов. Основными методами проверки правильности сложных систем являются имитационное моделирование, тестирование и проверка на модели. Методы имитационного моделирования и тестирования являются эффективными на ранних стадиях отладки системы и их результативность снижается, когда проектируемая система становится чище. Этот метод показал себя эффективным на начальном этапе отладки модели и позволил обнаружить ряд ошибок. Альтернативой методу имитационного моделирования является метод формальной верификации. Для этого могут быть использованы такие средства как темпоральные логики и алгоритмы стандартного анализа свойств множеств достижимых состояний для раскрашенных сетей Петри. Несомненным

преимуществом этих методов является то, что они автоматизированы. Темпоральные логики использовались для того, чтобы показать, что построенная модель адекватно отражает оригинальную спецификацию протокола. При этом мы не претендуем на то, что были проверены все свойства транспортного протокола. Основное препятствие состоит в том, что спецификация является достаточно громоздкой и носит неформальный характер, следовательно, нахождение и построение эквивалентных формальных требований на языке темпоральной логики достаточно затруднено.

Раздел 3.2 посвящен рассмотрению набора темпоральных формул логики ASK-CTL, которые позволяют верифицировать нашу модель предоставляемого протоколом сервиса. В этом разделе явно выписаны темпоральные формулы, которые проверяют корректность обработки пользовательских вызовов согласно стандарту RFC793 с учетом исправлений из RFC1122. Количество формул в наборе составляет 30 формул.

Любая формула из этого набора, имеет вид $\alpha \Rightarrow \mathcal{A}$, где α — это формула переходов, которая задает условие обработки некоторого пользовательского вызова, а $\mathcal{A}$ это темпоральная формула, которая проверяет обработку этого пользовательского вызова. Заметим, что α — это формула элементарных высказываний, которая в качестве условия включает как пользовательский вызов, так и условия (в частности, состояние локального соединения-адресата), при которых выражается его обработка. Можно сказать, что рассмотренная импликация задает частичную спецификацию обработки пользовательского вызова. Имея ввиду это, можно сформулировать следующую теорему:

Теорема 1 *Пусть* $\{\alpha_1 \Rightarrow \mathcal{A}_1, \ldots, \alpha_n \Rightarrow \mathcal{A}_n\}$ *— это набор свойств из раздела 3.2, тогда формула переходов логики ASK-CTL* $\Phi \equiv \wedge_{i=1}^{n}(\alpha_i \Rightarrow \mathcal{A}_i)$ *выражает полную спецификацию сервиса, предоставляемого протоколом TCP.*

Формула Φ может быть использована для формальной верификации модели следующим образом. Если эта формула выполняется на всех означивающих элементах системы, то система удовлетворяет специфи-

кации сервиса, предоставляемого протоколом TCP.

В разделе 3.3 предлагается подход для верификации случая, когда модель содержит несколько подсетей, моделирующих разные экземпляры протокола. При этом предлагается для каждой подсети, представляющей модель протокола TCP, запустить отдельную проверку алгоритма, представленного в предыдущем разделе. Так как количество таких подсетей в модели эксперимента конечно, то алгоритм будет выполняться также за конечное число шагов.

В разделе 3.4 описаны эксперименты по верификации процессов открытия и закрытия транспортного соединения. Первой фазой работы протокола TCP является фаза открытия соединения. Основной задачей этой фазы является синхронизация обеих сторон соединения. Установке соединения могут помешать следующие факторы:

1. потери синхронизирующих сегментов коммуникационной сетью;
2. вызовы программного процесса, которые могут мешать протоколу установить соединение;

В случае потерь сегментов производится их повторная передача, и, если сеть постоянно допускает потерю сегментов, то пользовательскому процессу в этом случае отказывается в установке соединения. Для того, чтобы исправить такое поведение коммуникационной сети, других средств, кроме повторной передачи, у транспортного протокола нет. В наших сценариях мы не будем моделировать потери синхронизирующих сегментов.

Вызовы программного процесса могут достаточно серьезно изменять поведение транспортного протокола. При этом на этапе установки соединения любая из сторон может, например, сбросить соединение и в этом случае получается полуоткрытое соединение. Нашей целью было исследовать, как влияют вызовы пользователя на процесс установки транспортного соединения.

В экспериментах были представлены две стороны транспортного соединения, которые пытаются синхронизироваться. При этом одна из сторон выступает в активной роли инициатора установки соединения, а вто-

рая — в пассивной роли стороны, которая принимает запрос на установку соединения. Каждой из сторон соединения пользовательский процесс отправляет вызовы, обработка которых зависит от состояния соединения. Причем, отправляются только такие вызовы, которые существенно изменяют состояние протокола, что влечет за собой, например, передачу сегментов удаленной стороне, сброс соединения и т.п. Множество таких вызовов было построено на основе анализа спецификации протокола RFC793 и RFC1122. Это множество является конечным для каждого из состояний, которое минует протокол в процессе установки соединения. Рассмотрим теперь, как изменяются состояния протокола в процессе выполнения такой модели:

1. пусть протокол находится в некотором состоянии $state$ и для него построено множество вызовов $u_1 \dots u_n$, которые при обработке существенно изменяют состояние $state$;
2. недетерминированно выбирается некоторый вызов $u_i \in u_1 \dots u_n$ и обрабатывается, переводя протокол в состояние $state'$;
3. если состояние протокла $state'$ не моделирует закрытое или синхронизированное соединение, переходим к шагу 1.

Необходимо еще отметить тот факт, что при установке соединения протокол обращает внимание на сам факт приема синхронизирущего сегмента, а не на то, когда он был принят. Таким образом, достаточно обойтись тривиальной моделью среды передачи данных, которая задерживает передачу сегментов на некоторое фиксированное время.

Для описанной системы была построена модель и исследовались ее свойства путем построения и анализа множества достижимых состояний. Построение производилось таким образом, чтобы в это множество попали только те состояния, которые относятся к фазе установки соединения. Множество достижимости для данной фазы состоит из 3792 достижимых разметок и 4156 дуг. Общее число терминальных разметок графа достижимых состояний составляет 425. Наибольший интерес для нас представляет то, к каким состояниям стремятся обе стороны соединения, а не состояния модели вцелом. Для этого из общего числа

тупиковых разметок модели были выделены тупиковые состояния обоих концов соединения. Выделение происходило по следующему критерию: два состояния модели являются эквивалентными, если они имеют идентичные разметки позиции, которая моделирует состояние контрольного блока соединения.

В результате получилось, что в совокупности обе стороны соединения стремятся к 6 состояниям. Одно из них задает случай, когда оба соединения сброшены и не существуют. Это происходит тогда, когда соединение (одна или обе стороны) сбрасывается с пользовательским процессом. В следующем случае у нас остается работающим одно пассивно открытое соединение. Этот случай допустим согласно спецификации протокола и происходит тогда, когда, например, активно открывающееся соединения сбрасывается, а пассивно открытое соединение ждет запроса от удаленной стороны. Оценка корректности такого случая зависит от того, какого поведения ожидает от транспортного протокола пользовательский процесс. В остальных 4-х случаях у нас в качестве тупиковых состояний получились два синхронизированных соединения, что является ожидаемым результатом процесса открытия соединения.

Построенное множество достижимых состояний не имеет циклов. Следовательно протокол открывает соединение за конечное число шагов. Было также проверено соответствие модели оригинальной спецификации протокола согласно подходу, описанному в разделе 3.2.

Аналогичный подход можно применить для анализа протокола в фазе закрытия соединения. Однако, для фазы передачи данных этот подход не является эффективным, так как работа протокола в процессе передачи данных зависит от коммуникационной сети, ее загрузки и других характеристик. Коммуникационная сеть может быть достаточно сложной и, соответственно, иметь очень большое множество достижимых состояний. Поэтому задача верификации транспортного протокола в процессе передачи данных является достаточно трудной.

Раздел 3.5 посвящен анализу работы алгоритма протокола ARTCP при множественных потерях сегментов коммуникационной сетью и представлению более эффективных алгоритмов работы. В работах И. В. Алек-

сеева и В. А. Соколова утверждается, что при потерях сегментов протокол ARTCP полагается на работу алгоритмов быстрой ретрансляции и повторной передачи по тайм-ауту. В случае потери одного сегмента существует вероятность того, что эта потеря будет эффективно восстановлена с помощью алгоритма быстрой ретрансляции. Однако когда теряется более одного сегмента, только первый сегмент может быть восстановлен с помощью механизма быстрой ретрансляции, а остальные восстанавливаются с помощью ретрансляции по тайм-ауту, что является неэффективным (см., напр. работы К. Фолла и С. Флойд). Поэтому для более эффективного восстановления от множественных потерь сегментов для протокола TCP вводятся алгоритмы NewReno как стандартный алгоритм (когда невозможно использовать выборочные подтверждения), и алгоритм SACK, который не является официальным стандартом (когда возможно использовать выборочные подтверждения).

Вскоре после представления протокола ARTCP был выпущен стандарт протокола RFC2988, который определяет процедуру установки и управления таймером повторной передачи. Этот документ рекомендует устанавливать таймер для первого сегмента в очереди повторной передачи и ретранслировать по срабатыванию таймера только его. Это обновляет алгоритм протокола ARTCP, так как в работах И. В. Алексеева и В. А. Соколова утверждается, что для протокола ARTCP от протокола TCP сохраняются алгоритмы определения значения тайм-аута и установки таймера повторной передачи.

В работе представлены два алгоритма восстановления от множественных потерь сегментов для протокола ARTCP — NewReno ARTCP, который используется в том случае, когда невозможно использовать выборочные подтверждения, и SACK ARTCP, когда возможно использовать выборочные подтверждения. Основная идея алгоритмов состоит в том, чтобы до тайм-аута попытаться определить, какие сегменты были потеряны и передать их в сеть. Несмотря на то, что для протокола TCP были разработаны алгоритмы для более эффективного восстановления от множественных потерь сегментов, отличающаяся специфика протоколов ARTCP и TCP не позволяет применять эти алгоритмы непосредственно

для протокола ARTCP.

В разделе 3.6 приводятся результаты модельных экспериментов по анализу производительности транспортных протоколов. Так как формализм раскрашенных сетей Петри позволяет выполнять построенные модели, то возможно применение методов имитационного моделирования для анализа свойств производительности. В экспериментах рассматривается модель простой сетевой инфраструктуры, которая дополнительно загружается синтетическим трафиком, обладающим свойством самоподобия. Это позволяет моделировать нагрузку на выбранную сетевую архитектуру. При различной загрузке была показана более эффективная работа протоколов SACK ARTCP и NewReno ARTCP, чем протоколов SACK TCP и NewReno TCP. Кроме того, были рассмотрены случаи снижения производительности алгоритма управления потоком ARTCP.

Основные выводы и результаты.

В рамках диссертации были получены следующие результаты.

- в работе представлена разработанная технология построения модели последней спецификации протокола TCP и методы ее модификации для моделирования будущих версий этого протокола;
- в работе были предложены подходы к верификации протокола TCP. В качестве примеров описана формальная верификация процессов открытия и закрытия транспортного соединения;
- разработаны алгоритмы для восстановления после множественных потерь сегментов для протокола ARTCP, которые учитывают несколько вариантов работы: с использованием механизма выборочных подтверждений (SACK ARTCP) или без них (NewReno ARTCP);
- предложены подходы к решению задачи анализа производительности коммуникационных транспортных протоколов. В ходе экспериментов была показана более эффективная работа протоколов SACK ARTCP и NewReno ARTCP, чем у протоколов SACK TCP и NewReno TCP. Кроме этого, были рассмотрены случаи снижения производительности алгоритма управления потоком ARTCP.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. *Dmitry Ju. Chaly, Valery A. Sokolov.* An Extensible Coloured Petri Net Model of a Transport Protocol for Packet Switched Networks // Proceedings of PaCT'2003, Springer–Verlag.— 2003.— № 2673.— p. 66-75 С. 58–69.

2. *Соколов В. А., Тимофеев Е. А., Чалый Д. Ю.* Моделирование, оптимизация и верификация транспортных протоколов // Труды Первой Всероссийской научной конференции МСО-2003, Москва.— 2003.— стр. 254–259.

3. *Чалый Д. Ю.* Моделирование протоколов TCP и ARTCP с помощью раскрашенных сетей Петри // Моделирование и анализ информационных систем, Ярославль.— 2003.— № 2— стр. 11–17.

4. *Соколов В. А., Чалый Д. Ю.* Методы исследования поведения транспортных протоколов в условия интенсивного сетевого трафика // Труды Международной конференции по вычислительной математике, Новосибирск.— 2004.— стр. 126–131.

5. *Чалый Д. Ю.* Моделирование транспортных протоколов коммуникационной сети Интернет // Труды 3-й междисциплинарной конференции НБАТТ-21, Петрозаводск.— 2004.— стр. 76–77.

6. *Алексеев И. В., Соколов В. А., Чалый Д. Ю.* Моделирование и анализ транспортных протоколов в информационных сетях // ЯрГУ им. П.Г. Демидова, Ярославль.— 2004.— 262 с.

7. *Чалый Д. Ю.* Анализ и верификация моделей транспортных протоколов коммуникационной сети Интернет // Труды XVI Международной научно-технической конференции «Математические методы и информационные технологии в экономике, социологии и образовании», Пенза.— 2005.— стр. 199–201.

Личный вклад автора. Все включенные в диссертацию результаты по построению, анализу, а также модификации моделей семейства транспортных протоколов в терминах раскрашенных сетей Петри получены автором лично. Разработанные алгоритмы для более эффективного восстановления от множественных потерь сегментов для протокола ARTCP были получены самостоятельно.